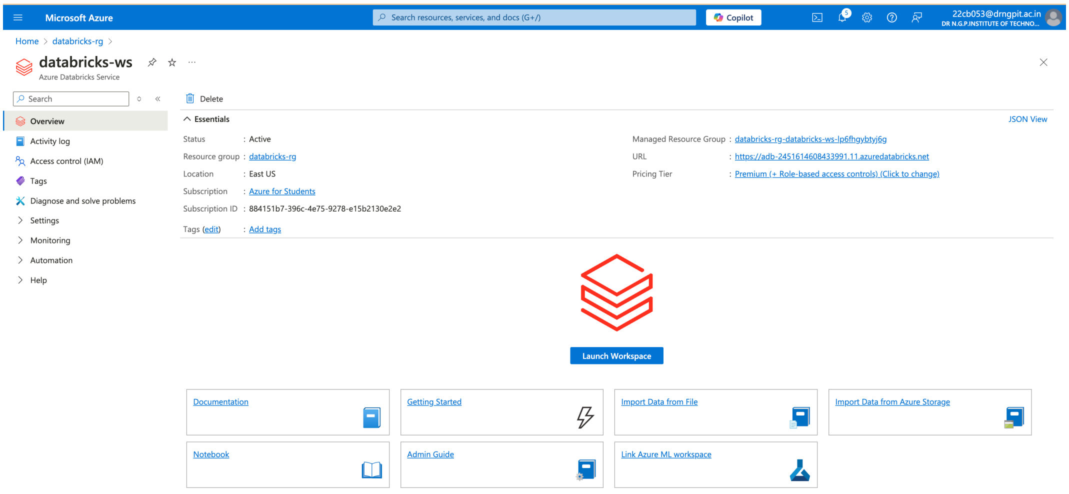
Step1 : Setting database containers and Azure data bricks



databricks-ws

Azure Databricks Service

Essentials

Status : Active

Resource group : databricks-rg

Location : East US

Subscription : Azure for Students

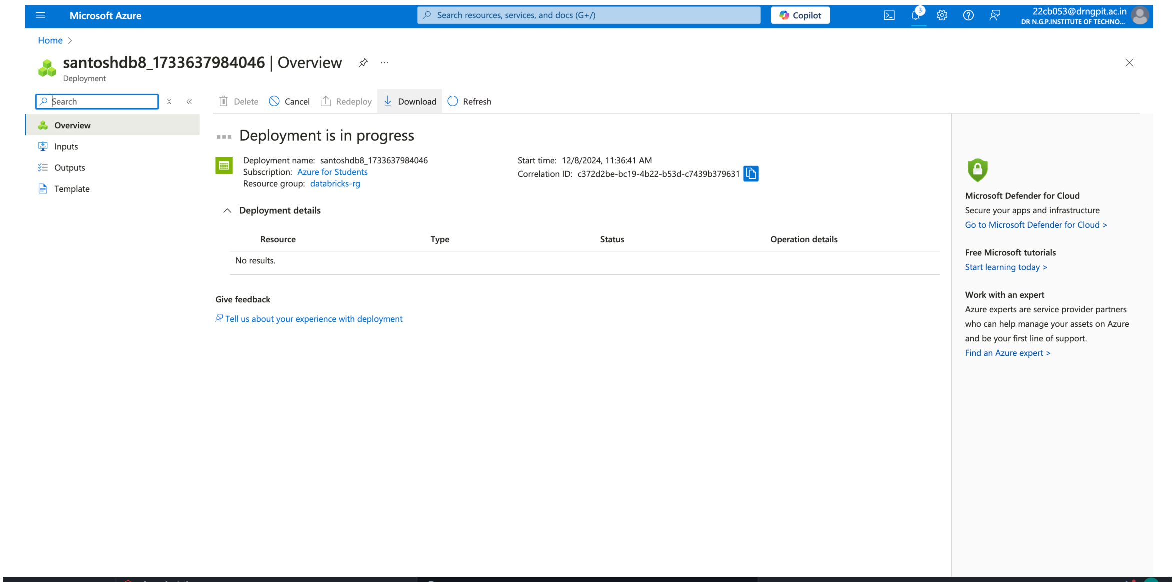
Subscription ID : 884151b7-396c-4e75-9278-e15b2130e2e2

Tags (edit) : Add tags

Managed Resource Group : databricks-rg-databricks-ws-lp6fhgybtyj6g

URL : https://adb-2451614608433991.11.azuredatabricks.net

Pricing Tier : Premium (+ Role-based access controls) (Click to change)



santoshdb8_1733637984046 | Overview

Deployment

Deployment is in progress

Deployment name: santoshdb8_1733637984046
Subscription: Azure for Students
Resource group: databricks-rg

Start time: 12/8/2024, 11:36:41 AM
Correlation ID: c372d2be-bc19-4b22-b53d-c7439b379631

Deployment details

Resource	Type	Status	Operation details
No results.			

Give feedback

Tell us about your experience with deployment

Microsoft Defender for Cloud

Secure your apps and infrastructure

Go to Microsoft Defender for Cloud >

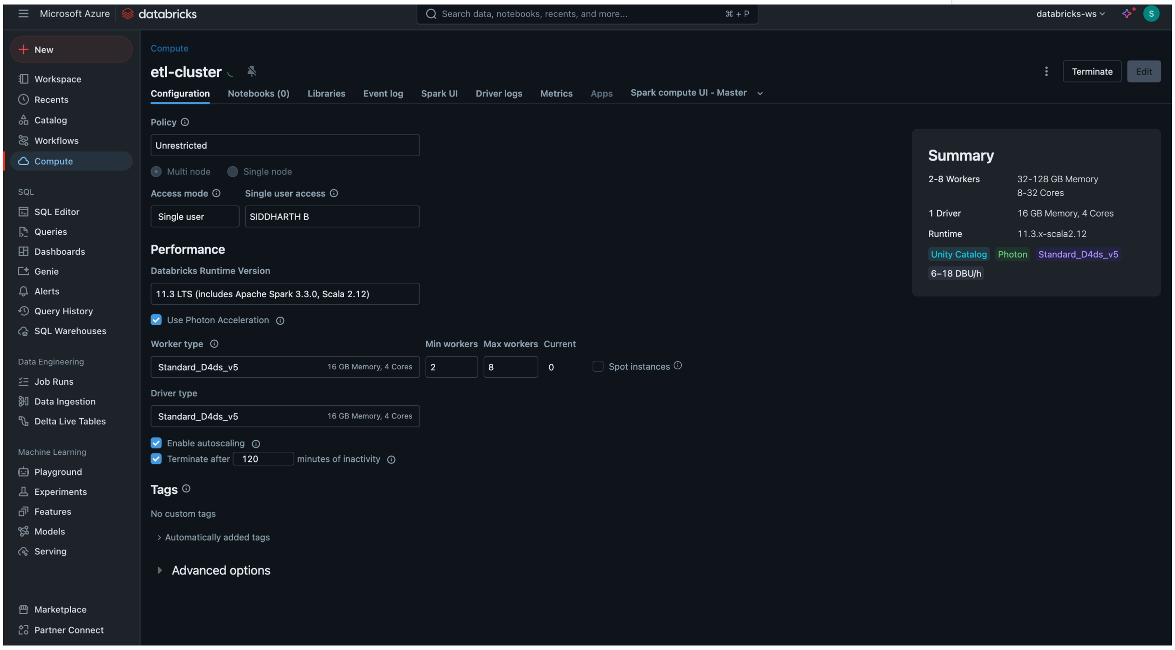
Free Microsoft tutorials

Start learning today >

Work with an expert

Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.

Find an Azure expert >



etl-cluster

Policy: Unrestricted

● Multi node ○ Single node

Access mode: Single user

Single user access: SIDDHARTH B

Performance

Databricks Runtime Version: 11.3 LTS (includes Apache Spark 3.3.0, Scala 2.12)

☑ Use Photon Acceleration

Worker type: Standard_D4ds_v5 16 GB Memory, 4 Cores

Min workers: 2 Max workers: 8 Current: 0

□ Spot instances

Driver type: Standard_D4ds_v5 16 GB Memory, 4 Cores

☑ Enable autoscaling

☑ Terminate after 120 minutes of inactivity

Tags

No custom tags

› Automatically added tags

Advanced options

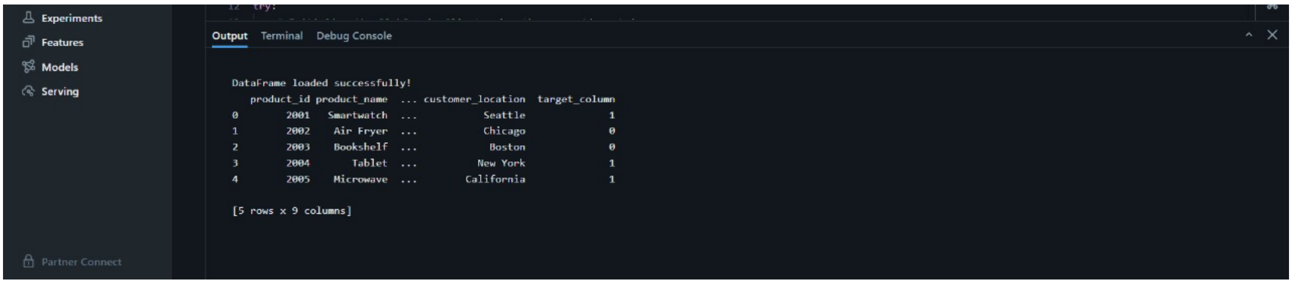
Summary

2-8 Workers	32-128 GB Memory 8-32 Cores
1 Driver	16 GB Memory, 4 Cores
Runtime	11.3.x-scala2.12

Unity Catalog Photon Standard_D4ds_v5

6–18 DBU/h

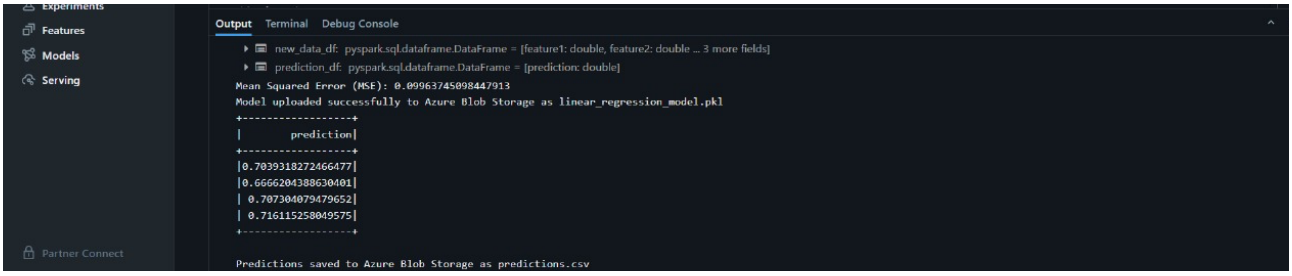
Step 2: Run the python code in Azure blob notebook (Load large dataset, preprocess, Implement ml):



```
try:
```

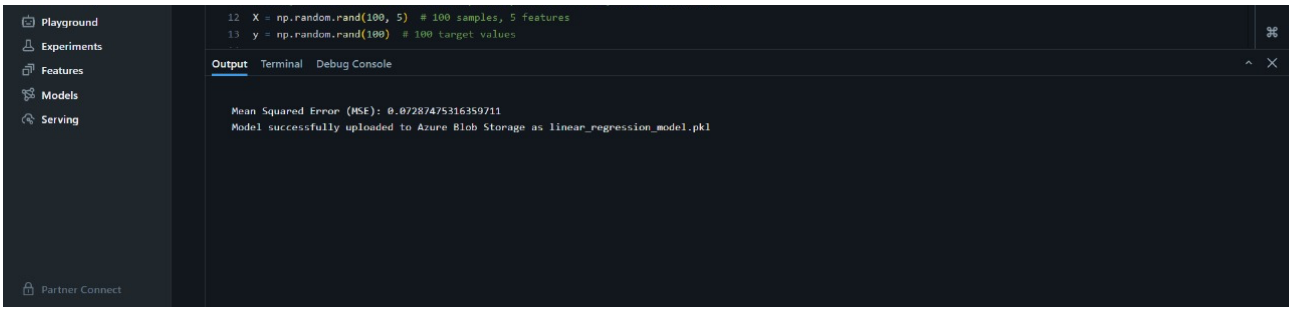
```
DataFrame loaded successfully!
   product_id product_name  ... customer_location  target_column
0        2001   Smartwatch  ...           Seattle              1
1        2002    Air Fryer  ...           Chicago              0
2        2003    Bookshelf  ...            Boston              0
3        2004       Tablet  ...          New York              1
4        2005    Microwave  ...        California              1

[5 rows x 9 columns]
```



```
▸ new_data_df: pyspark.sql.dataframe.DataFrame = [feature1: double, feature2: double ... 3 more fields]
▸ prediction_df: pyspark.sql.dataframe.DataFrame = [prediction: double]
Mean Squared Error (MSE): 0.09963745098447913
Model uploaded successfully to Azure Blob Storage as linear_regression_model.pkl
+------------------+
|        prediction|
+------------------+
|0.7039318272466477|
|0.6666204388630401|
| 0.707304079479652|
| 0.716115258049575|
+------------------+

Predictions saved to Azure Blob Storage as predictions.csv
```



```
X = np.random.rand(100, 5)  # 100 samples, 5 features
y = np.random.rand(100)  # 100 target values
```

```
Mean Squared Error (MSE): 0.07287475316359711
Model successfully uploaded to Azure Blob Storage as linear_regression_model.pkl
```